

## **REMARKS**

In the Official Action mailed on **30 January 2006**, the Examiner reviewed claims 1-20. Claims 1-5, 10-11, 13, and 18-20 are rejected under 35 U.S.C. § 102(b) as being anticipated by “Speculative Lock Elision: Enabling Highly Concurrent Multithreaded Execution” by Ravi Rajwar and James R. Goodman (hereinafter “Rajwar”). Claim 12 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Rajwar in view of common art. Claims 2-5 and 14-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rajwar in view of “Enhancing Software Reliability with Speculative Threads” by Jeffrey Oplinger and Monica S. Lam (hereinafter “Lam”). The examiner further objected to the title as being not descriptive.

### **Objections**

The title was objected to for being not descriptive. Applicant has amended the title to indicate clearly the invention to which the claims are directed.

### **Rejections under 35 U.S.C. § 102(b) and 35 U.S.C. § 103(a)**

Independent claims 1 and 13 were rejected as being anticipated by Rajwar. Applicant respectfully points out that Rajwar discloses “speculative lock elision,” a **hardware-based** mechanism for dynamically predicting unnecessary lock operations and eliding such operations. *See* Rajwar at 295, col. 1, lines 29-35. The Rajwar mechanism is implemented “entirely in microarchitecture, **without instruction set support** and without system-level modifications and is **transparent to programmers.**” *See* Rajwar at 295, col. 2, lines 16-19 (emphasis added). According to Rajwar, a programmer has no control over the lock prediction and elision mechanism. *See* Rajwar at 295, col. 2, lines 19-23.

In contrast, the present invention provides a special “start transactional execution” (STE) **instruction**, which can be explicitly executed before a process executes a critical code section. *See* instant application at 12, line 10. The instant

application teaches away from Rajwar, because to implement the STE instruction, the processor's **instruction set is augmented** to include the STE instruction. *See* instant application at 12, lines 19-22. The Rajwar system, conversely, does not modify the instruction set at all. The STE instruction, as taught in the instant application, performs a series of operations before the process starts executing the critical code section. *See* instant application at 14, lines 1-10. Furthermore, the STE instruction **is not part of the critical code section**, and is executed prior to the execution of the critical code section. *See* instant application at 12, line 27; at 14, lines 16-18. This mechanism allows a programmer or compiler to explicitly control the speculative execution process, which is not available in the Rajwar system.

Accordingly, Applicant has amended independent claims 1, 13, and 25 to clarify that the present invention provides a "start transactional execution" instruction that is executed before a process executes a critical code section. These amendments find support in FIGs. 3 and 4, and in paragraphs [0059] to [0061] of the instant application.

Hence, Applicant respectfully submits that independent claims 1, 13, and 25 as presently amended are in condition for allowance. Applicant also submits that claims 2-12, which depend upon claim 1, and claims 14-24, which depend upon claim 13, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

## CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By

  
Edward J. Grundler  
Registration No. 47,615

Date: April 26, 2006

Edward J. Grundler  
PARK, VAUGHAN & FLEMING LLP  
2820 Fifth Street  
Davis, CA 95616-7759  
Tel: (530) 759-1663  
FAX: (530) 759-1665  
Email: [edward@parklegal.com](mailto:edward@parklegal.com)